

27장. 배열

27.1 배열이란?

1. 배열(array)은 여러 개의 값을 순차적으로 나열한 자료구조다.
2. 배열이 가지고 있는 값을 **요소(element)**,
요소 자신의 위치를 나타내는 0 이상의 정수인 **인덱스(index)**를 갖는다.
3. 자바스크립트에 배열이라는 타입은 존재하지 않는다. 배열은 객체 타입이다.

```
typeof arr // object
```

4. 배열과 일반 객체

Aa 구분	≡ 객체	≡ 배열
구조	프로퍼티 키와 프로퍼티 값	인덱스와 요소
값의 참조	프로퍼티 키	인덱스
값의 순서	X	O
length 프로퍼티	X	O

- 둘의 가장 명확한 차이는 “값의 순서”, “length 프로퍼티”

27.2 자바스크립트 배열은 배열이 아니다

1. **밀집 배열(dense array)**

: 배열의 요소는 하나의 데이터 타입으로 통일되어 있으며 서로 연속적으로 인접해 있다.

2. 인덱스를 통해 단 한번의 연산으로 임의의 요소에 접근

: 임의 접근(random access), 시간 복잡도 O(1)

정렬되지 않은 배열에서 특정한 요소를 검색하는 경우,
특정 요소를 발견할 때까지 처음부터 차례대로 검색
: 선형 검색(linear search), 시간 복잡도 $O(n)$

3. 배열의 요소를 위한 각각 메모리 공간은 동일한 크기 아니어도 되며,
연속적으로 이어져 있지 않을 수도 있음.

: **희소 배열**(sparse array)

- 자바스크립트 배열은 일반적인 배열의 동작을 흉내 낸 특수한 객체다.

4. 자바스크립트에서 배열이 일반 객체보다 약 2배 정도 빠르다.

27.3 length 프로퍼티와 희소 배열

1. 현재 length 프로퍼티 값보다 작은 숫자 값을 할당하면 배열의 길이가 줄어든다.
2. 현재 length 값보다 더 큰 숫자 값을 할당하는 경우, length 프로퍼티 값은 변경되지만 실제로 배열의 길이가 늘어나지는 않는다.
3. 이처럼 배열 요소가 연속적으로 위치하지 않고 일부가 비어 있는 배열을 **희소 배열**이라 함.
4. **희소 배열은 length와 배열 요소의 개수가 일치하지 않는다.**
희소 배열의 length는 희소 배열의 실제 요소 개수보다 언제나 크다.

27.4 배열 생성

27.4.1 배열 리터럴

1. 가장 일반적이고 간편한 배열 생성 방식이다.

27.4.2 Array 생성자 함수

1. Array 생성자 함수를 통해 배열을 생성한다.
전달된 인수의 개수에 따라 다르게 동작하므로 주의가 필요.

2. `new` 연산자와 함께 호출하지 않더라도, 즉 일반 함수로서 호출해도 배열을 생성하는 생성자 함수로 동작한다.
이는 생성자 함수 내부에서 `new.target`을 확인하기 때문.

```
Array(1, 2, 3); // [ 1, 2, 3 ]
```

27.4.3 Array.of

1. 전달된 인수를 요소를 갖는 배열을 생성

27.4.4 Array.from

1. 유사 배열 객체 or 이터러블 객체를 인수로 전달받아 배열로 변환하여 반환한다.

```
// 유사 배열 객체를 변환하여 배열을 생성
Array.from({ length: 2, 0: 'a', 1: 'b' }); // ['a', 'b']

// 이터러블을 변환하여 배열을 생성. 문자열은 이터러블이다.
Array.from('Hello'); // ['H', 'e', 'l', 'l', 'o']
```

27.5 배열 요소의 참조

1. 존재하지 않는 요소에 접근하면 `undefined`가 반환

27.6 배열 요소의 추가와 갱신

1. 존재하지 않는 인덱스를 사용해 값을 할당하면 새로운 요소가 추가된다.
이때 `length` 프로퍼티 값은 자동 갱신됨.
2. 정수 이외의 값을 인덱스처럼 사용하면 요소가 생성되는 것이 아니라, 프로퍼티가 생성됨.

27.7 배열 요소의 삭제

1. 배열은 사실 객체이기 때문에 `delete` 연산자 사용 가능

2. 회소 배열을 만들게 될 수 있으므로 delete 연산자는 사용하지 않는 것이 좋다. 🙅

27.8

27.9